

Building a Book Scanner Rig

There are a lot of options for building a rig. You can build an established design from scratch, or forge out on your own and make something completely new.

Cardboard Scanner (<http://www.instructables.com/id/Bargain-Price-Book-Scanner-From-A-Cardboard-Box/>)

The cheapest option is to build your scanner from a cardboard box. It is also a relatively easy rig to build. The quality may suffer, but in some situations that it a good trade-off to make.



Hardware Store Scanner (<http://www.diybookscanner.org/forum/viewtopic.php?f=1&t=333>)

The hardware store scanner is a more durable alternative to the cardboard scanner. No custom parts are needed and the only tools you need are standard woodworking power tools. If you walk into a hardware store, you can walk out with everything you need to build this scanner. You can expect to get better scans with this rig than the cardboard scanner.



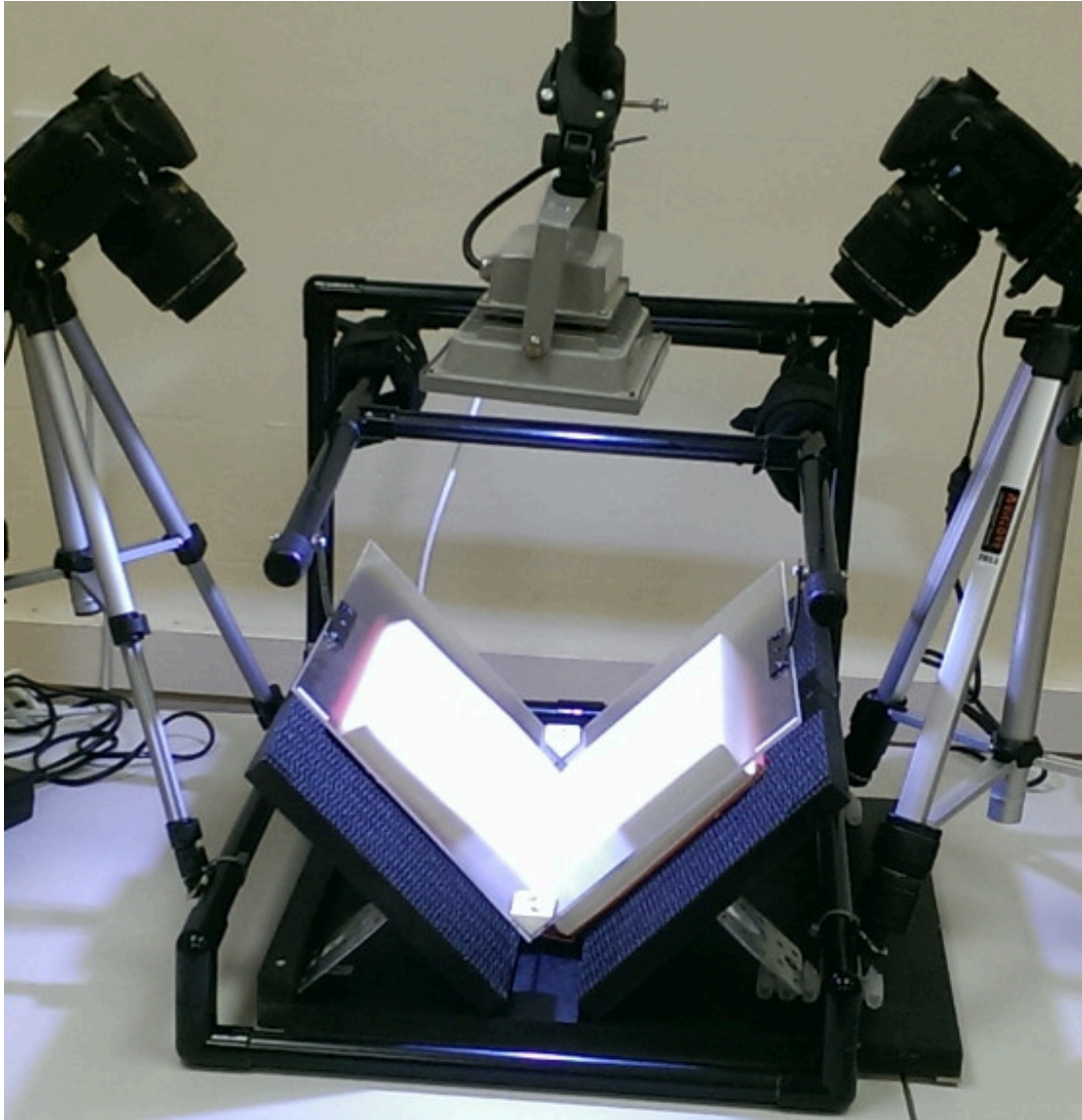
Archivist from Scratch (<http://diybookscanner.org/archivist>)

The Archivist is the culmination of 6 years of scanner design. It allows you to make high quality scans quickly. The lighting is even with no glare and it is also great for color scans and glossy pictures. Daniel Reetz released the design of it into the public domain so you can look through his complete design guide and build one yourself from scratch.



Plastic Tubing Scanner (<http://www.instructables.com/id/Book-Scanner-Low-cost-easy-to-make-1000-pages-an-h/>)

In 2013, David Landin posted about a new design using lightweight PVC plastic tubing as the main construction material. He has produced videos describing how to construct one for yourself and many people have. Some people have also worked to adapt his design for handling very large books which is an unsolved problem for many other scanner rigs. The forum thread is here (<http://diybookscanner.org/forum/viewtopic.php?f=14&t=2914>).





Other Designs (designs.html)

The idea of book scanning has captured the imaginations of people all over the world. Dozens of people have posted design ideas, photos of their projects, and even complete plans in our forum (<http://diybookscanner.org/forum>). We have put many of the scanner photographs into a gallery (designs.html) which links back to the original forum threads. Browse through some of the designs and if you find one you like, start building it.

Make Something New (<http://diybookscanner.org/forum>)

Maybe none of the current designs fit your exact requirements. Or maybe you have a new idea that will change everything. If you want to pursue a new design, please chronicle your exploration on the forum (<http://diybookscanner.org/forum>). Other members of the community can provide encouragement and advice. Make sure to read through the Archivist Design Guide (<http://diybookscanner.org/archivist>) for a discussion of the various aspects of scanner design.

Picking Cameras

There are a wide variety of cameras that you can scan with. If you plan on using Pi Scan (<http://github.com/Tenrec-Builders/pi-scan>) to control your cameras, then you should use Canon PowerShot ELPH 160 cameras. But if you are using some other setup, then here are some general guidelines for choosing cameras.

Selecting the right camera is really important. We have years and years of debate on the topic. No question gets asked more often, and so nobody has thought about this more than the DIY book scanning community. And we have a three step process for you to figure it out.

Step 1. How many megapixels do you need?

A. Measure at the books you intend to scan. Aim for the largest average size (don't choose the largest outliers). For example, most textbooks are around 9 x 11 in (22.86cm x 27.94cm).

B. Now multiply that size by the PPI (pixels per inch) that you intend to capture. 300 is a safe minimum, though you can't go wrong by capturing higher than that. So, in our example - $9 \times 300 = 2700$. $11 \times 300 = 3300$. We need an image that's at least $2700 \times 3300 = 8910000$ pixels, or about 9 megapixels. Now, that's if you used every pixel perfectly to capture every part of the page, which NEVER happens. So to be safe, add 20-30% for wasted pixels. In this case, that makes 12 megapixels the minimum to get at least 300PPI capture.

Step 2. How much control do you need?

If you're just scanning one book, or you're scanning a book for its information content only (as opposed to trying to capture the actual physical appearance of the book), you don't need very good captures. If the lighting changes, or the camera settings change from shot-to-shot, you'll still get some kind of result. However, the more perfectly you want to capture the book, and the more pages you want to capture, the more control you need. So assuming you want to do a good job and care about more than just the raw text on any page, you need a camera that lets you control the following:

1. Shutter speed
2. White balance
3. Aperture
4. ISO
5. Flash on/off
6. Any custom image processing (sharpening, color enhancements, etc)
7. Focus (ideally being able to lock focus)
8. Exposure compensation
9. Zoom

Most DSLRs allow for all this kind of control; for compact cameras only Canon Powershot cameras that are capable of running CHDK (<http://chdk.wikia.com>) give you control over all these parameters. To see if a camera is capable of running CHDK, you can check here (<http://www.mighty-hoernsche.de/>).

One more factor to consider: ideally you want to run the cameras from an AC adapter instead of batteries. Check availability of these accessories.

Step 3. How much money do you have?

If you have a healthy budget, just buy DSLR cameras and use those. Buy the highest resolution you can afford, and try the "kit lens" that comes with the camera body as a starting place (they usually cost only \$50-100 over the price of the camera body alone and perform reasonably well).

If you're on a budget, the aforementioned Canon compact cameras can often be purchased for as little as \$75 USD each, and, with CHDK, produce incredibly high-quality images. They are by far the best "bang for the buck" - which is what DIY Scanning is all about.

CHDK and Canon Cameras

Most cheap compact cameras do not have a software interface. They can be controlled only by manual or mechanical triggering. But a team of volunteers has developed software which can allow Canon compact cameras to be controlled and configured remotely. This software is called CHDK (<http://chdk.wikia.com>).

CHDK is loaded onto an SD card which is then inserted into the camera. When the camera starts up, CHDK is run automatically. Since CHDK never makes any permanent changes to the camera, you can always just remove the special CHDK SD card to run the camera normally.

CHDK is an essential pre-requisite to the software controllers listed below. The controllers run on a PC or Raspberry Pi and communicate with the CHDK software running on the cameras over USB. CHDK provides many enhanced capabilities, including the ability to configure the camera over USB, capture photographs, and then transfer the resulting images over USB to the controller.

Because CHDK is so useful and there is no equivalent for other kinds of cheap point and shoot cameras, most users in the forums use Canon cameras in their rigs. When using other kinds of cheap cameras, the only control option is some kind of mechanical or manual triggering.

Controlling the Cameras

The first task when digitizing books is capturing an image of each page and then putting those images in a convenient place. There are a few ways you can go about this task.

Pi Scan (<http://github.com/Tenrec-Builders/pi-scan>)

Pi Scan is a scanning appliance which runs on a Raspberry Pi 2. It supports Canon PowerShot A2500 and Canon PowerShot ELPH 160 cameras. You plug your cameras into it along with a monitor, mouse, optional keyboard, and an external disk. It configures and triggers both cameras, saving the images to external storage. When you are done, your external disk will have a folder full of image scans.

Spreads (<http://www.diybookscanner.org/forum/viewforum.php?f=35>)

Spreads can also run on a Raspberry Pi, though it works on many flavors of Linux as well. It works with many CHDK-compatible cameras. You control Spreads over the network via a web interface. It configures and controls the cameras and keeps them in named 'Workflows'. When a workflow is complete, you can download a compressed archive of the images or save them onto an external drive.

TwoCamControl (<http://www.diybookscanner.org/forum/viewtopic.php?f=20&t=3082>)

If you have a Windows machine, you can run the TwoCamControl script directly on your PC. It works with many CHDK-compatible cameras. This means that you don't have to buy any extra hardware or worry about interfacing with it. Just specify a directory and the images will be saved there.

Mechanical Trigger (<http://diybookscanner.org/forum/search.php?keywords=bicycle+brake>)

If none of the software options works for you, there is still hope for an ergonomic solution. You just need a way to replicate a finger pushing against the shutter button on your camera. The most common way to do this is by repurposing bicycle brake systems. Bicycle brakes work by pneumatic pressure. When you squeeze the lever, it increases the pressure in a tube. This pressure can be used to push a piston positioned directly over the shutter button of a camera.

After capturing images using a mechanical trigger, you will still need to copy the images off of the cameras and onto your computer for post-processing. If possible, use a data cable to do this. Otherwise you may have to unmount your cameras to get access to the physical cards inside.

Manual Trigger

When all else fails, you can trigger cameras by hand. They were designed to be operated this way after all. But doing this hundreds or thousands of times can be painful. It is hard to design a rig to allow ergonomic operation when triggering cameras. And even worse, by touching the camera whenever you take a picture, you are also introducing a vibration into the system just at the time when you want to take a photograph.

Images to eBooks

After capture, you will have a folder full of images. Turning those images into an eBook is called 'post-processing'. What steps this actually entails depends on your needs. Some people want to compress things down as much as possible and extract the text of the book using OCR. Others just want to crop each image to the page and bind them into a PDF. A free book called E-Book Enlightenment (<https://en.flossmanuals.net/e-book-enlightenment/>) has sections about how to make e-books. There are also a number of software tools to help you perform these tasks. Here are a few:

Scan Tailor (<http://www.diybookscanner.org/forum/viewforum.php?f=21>)

A full-featured tool which can do many kinds of manipulation. Including image rotation, crop to content, deskewing, dewarping, and binarization.



Scan Tailor tutorial

Joseph Artsimovich

20:58

Scan Tailor tutorial (<https://vimeo.com/12524529>) from Joseph Artsimovich (<https://vimeo.com/user4035740>) on Vimeo (<https://vimeo.com>).

Book Scan Wizard (<http://www.diybookscanner.org/forum/viewforum.php?f=22>)

A postprocessing tool providing rotation, cropping, dewarping and more. It is oriented towards power users.

BookScanWizard demo



Commercial Options

Abbyy Finereader, Adobe Acrobat, and Omnipage are all paid alternatives. You will need to review their literature to see if they are appropriate for your requirements.

Questions? Ideas? Join us in the Forum. (<http://diybookscanner.org/forum>)